UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/240,406	01/29/1999	JOSEPH P. FERNANDO	777.229US1	7291

7590 10/27/2003

STEVEN J. ROCCI
WOODCOCK WASHBURN KURTZ& MACKIEWICZ & NORRIS LLP
ONE LIBERTY PLACE-46TH FLOOR
PHILADELPHIA, PA 19103

EXAMINER	
LAO, SUE X	
ART UNIT	**PAPER NUMBER**
2126	

DATE MAILED: 10/27/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

## *Office Action Summary*	Application No. 09/240,406	Applicant(s) Fernando et al	
	Examiner S. Lao	Art Unit 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *Aug 1, 2003* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *16-21 and 27-44* is/are pending in the application.

4a) Of the above, claim(s) ______ is/are withdrawn from consideration.

5)☐ Claim(s) ______ is/are allowed.

6)☒ Claim(s) *16-21 and 27-44* is/are rejected.

7)☐ Claim(s) ______ is/are objected to.

8)☐ Claims ______ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on ______ is/are a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on ______ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some* c)☐ None of:

1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ______ .
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

a)☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s). ______

4) ☐ Interview Summary (PTO-413) Paper No(s). ______

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other:

DETAILED ACTION

1. Claims 16-21, 27-44 are pending. This action is in response to the amendment filed 6/30/2003 and an RCE filed 8/1/2003. Applicant has amended claims 16, 27, 29, 34, 42 and 44 and canceled claims 11-15 and 22-26.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

3. Claims 16, 18-20, 27, 28, 42, 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baxter et al (U S Pat. 6,289,500) in view of Sonderegger (U S Pat. 5,761,499).

As to claims 16 and 42, Baxter teaches a system (IBM San Francisco framework) for extending functionality (to include domain-specific functions) of a class object (ExtensibleItem class which is domain-neutral), comprising: processing unit (110); system memory (120); system bus (160); computer-readable medium (155); and an extensible object model (San Francisco Framework) executed from, wherein the extensible object model creates (DomainItemCreator) an extension object (DomainExtension object) from an extension package (DomainExtension class which implements domain-specific functions) when a requested functionality (domain-specific functions) is not inherent in the class object [it is noted that the domain-neutral extensible items/objects do not provide domain-specific functions], and wherein the extension object extends the class object to provide the requested functionality (provide domain-specific functions to domain-neutral extensible items/objects). See col. 8, lines 45-61; col. 10, line 19 - col. 11, line 67.

Baxter does not explicitly teach the extensible object model determines whether a requested functionality is inherent in the class object.

Sonderergger teaches extending functionality of a class object (COM object), including determining whether a requested functionality is inherent (registered in the registry) in the class object (check the registry for availability of the desired COM component, col. 10, lines 31-38). If the requested functionality is not inherent in the class object (the desired COM component not registered), an extension package is located (query directory services and databases listing unregistered COM/OLE components on component server, col. 10, line 39 - col. 11, line 10) to create an extension object (COM interface IClassFactory). Given the teaching of Sonderergger, it would have been obvious to include a step to determine whether a requested functionality is inherent in the class object. A motivation to combine the teaching of Baxter with Sonderergger would have been to use extension packages located on other nodes of the network (Sonderergger, col. 11, lines 52-65) to provide more resources of extensions.

As to claim 18, 43, Baxter teaches registering the extension package in an extension database (persistent collection, col. 11, lines 16-22).

As to claim 19, Baxter teaches store the extension object in system memory (dynamic virtual function table) when the corresponding extension is first referenced (col. 7, lines 20-29.

As to claim 20, Baxter teaches creating an extension provider object (factory ExtensibleItemSpecialFactory) and create the extension object from the extension provider object (create extensions). See col. 11, lines 1-67.

As to claim 27, Baxter teaches extensible object (ExtensibleItem), extension database (persistent collection) having an entry for an extension (extension for a particular domain, ie, of type DomainInterface) for the extensible object; extension package (DomainExtension class and DomainItemCreator) having an interface for obtaining (DomainItemCreator) an extension object (DomainExtension) that provides the extension for the extensible object. See col. 10, line 19 - col. 11, line 67. Note discussion of claim 16 for the determining step and a motivation to combine the teachings of Baxter / Graser / IBM San Francisco framework with Sonderergger.

As to claim 28, Baxter teaches a call to the interface in the extension package (client call). See col. 10, lines 64-67.

4. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Graser et al (U S Pat. 6,275,979) in view of Sonderegger (U S Pat. 5,761,499).

As to claim 34, Graser teaches a method (San Francisco framework) for extending functionality (support additional method) of a class object (ExtensibleItem) in a run-time environment (at run-time), comprising: receiving a request (invokeMethod()) from an application (client) for functionality that is not inherent in the class object [it is noted that ExtensibleItem has no domain-specific information]; determining if the functionality is available (locate the method name via method table) in a first extension object (Extension1); and directing the request to the functionality in a second extension object (Extension2), when the functionality is not available in the first extension object [It is noted that when looking for arb() method, Extension2 will be returned instead of Extension1]. See col. 5, line 58 - col. 6, line 8; col. 6, line 30 - col. 7, line 18; col. 9, lines 2-9; col. 11, lines 6-10. Note discussion of claim 16 for the determining step and a motivation to combine the teachings of Graser / IBM San Francisco framework with Sonderergger.

5. Claims 17, 29-33, 35-41, 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over the IBM San Francisco framework as disclosed by Baxter et al and Graser et al in view of Sonderegger.

It is noted that both Baxter and Graser describe the run-time operations of the same IBM San Francisco framework, emphasizing on different aspects. The combination of Baxter and Graser provides a more complete picture of the San Francisco framework. Therefore, it would have been obvious to combine the teachings to provide enhancement in various aspects.

As to claim 17, IBM San Francisco framework provides (Graser) notifying the extensible object when the extension object is deleted (previous extension overridden and deleted, col. 7, lines 18-53).

As to claim 29, the IBM San Francisco framework provides (Baxter) a method for extending functionality (new domain extension) of a class object (ExtensibleItem) in a run-time environment (San Francisco framework), comprising: receiving a request from an application (client invokes) for functionality that is not inherent in the class object (new domain extension needed); determining if the functionality is available in a first extension object (locate special factory ExtensibleItemSpecialFactory); obtaining an extension package (classes, collections and factories) having computer-executable instructions associated with the extension object functionality (extension of type DomainInterface), wherein the extension package proffers an extension provider object (special factory) when the functionality is requested; specifying parameters (pass domain parameters) to the extension provider object to create a second extension object (create extension object via ExtensibleItemFactory). See col. 11, lines 6-67. The IBM San Francisco framework also provides (Graser) a step for directing the invocation to the second extension object (Extension2 which implements the requested arb()) after the second extension object has been created. See col. 5, line 58 - col. 6, line 8; col. 6, line 30 - col. 7, line 18; col. 9, lines 2-9; col. 11, lines 6-10. Note discussion of claim 16 for the determining step and a motivation to combine the teachings of Baxter / Graser / IBM San Francisco framework with Sonderergger.

As to claim 30, note discussion of claim 18.

As to claims 31, 36, note discussion of claim 27 (Baxter) for storing the extension package in an extension database.

As to claims 32, 40, San Francisco framework provides (Graser) searching for an entry associated with the functionality (col. 6, lines 9-41).

As to claims 33, 41, San Francisco framework provides (Graser) creating the second extension object when the extended functionality is first referenced (create new extension and add to method table), and locating (look up method name) the second extension object when the extended functionality is subsequently referenced (col. 6, lines 9-65).

As to claim 35, note discussion of claim 29 for obtaining an extension package.

As to claim 37, note discussion of claim 18. register the extension package in an extension database stored on.

As to claims 38 and 39, note discussion of claim 20.

As to claim 44, the IBM San Francisco framework provides (Baxter) a method for extending functionality (new domain extension) of a class object (ExtensibleItem which is domain-neutral), comprising: invoking (client invokes) a functionality that is not inherent in the class object (new domain extension); determining if the invoked functionality is available in a first extension object (look for special factory ExtensibleItemSpecialFactory that should be used); creating a second extension object (use standard factory ExtensibleItemFactory) when the invoked functionality is not available in the first extension object (otherwise use the standard factory). See col. 11, lines 1-11, 39- 50. The IBM San Francisco framework also provides (Graser) a step for directing the invocation to the second extension object (Extension2 which implements the requested arb()) after the second extension object has been created. See col. 5, line 58 - col. 6, line 8; col. 6, line 30 - col. 7, line 18; col. 9, lines 2-9; col. 11, lines 6-10. Note discussion of claim 16 for the determining step and a motivation to combine the teachings of Baxter / Graser / IBM San Francisco framework with Sonderergger.

6. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Baxter et al in view of Sonderergger as applied to claim 16 and further in view of Schmidt et al ("An Object-Oriented Framework for Developing Network Server Daemons").

As to claim 21, Schmidt teaches framework based software architecture (service configuration), including creating an event filtering and sourcing object (event handler) to handle events (events) generated by an extension object (service object). See pages 7-8, section 4.1. Therefore, it would have been obvious to create an event filtering and sourcing object in Baxter to handle events generated by an extension object. In so doing, configuring different types of I/O events from a client would have been simplified with the class library (page 6, section 3.2.3).

7. Applicant's arguments filed 6/30/2003 have been considered but are moot in view of the new ground(s) of rejection.

Sonderergger is cited to teach determining whether a requested functionality is inherent in a class object before using an extension package to provide the requested functionality. Refer to discussion of claim 16 for detailed discussion.

Regarding the argued "a previously created extension object from one vendor's application is made available to another vendor's application" (page 6, 2nd para.) and "extensions that are accessible through late or ID binding" (page 7, 1st para.) are not brought out by the claim language.

Regarding the argument that Baxter is an architecture that enables customization of an application, whereas the present invention provides "extensions that do not have to be present when an application is developed" (page 8, 2nd para.), the architecture of Baxter is an object infrastructure which provides dynamic run-time extension of object support. See Baxter, col. 5, line 61 - col. 6, line 15. In other words, it is an extensible object model. Further, as disclosed, applicant's extensible object model refers to an application. See application as filed, page 9, lines 8-9. Regarding the "extensions that do not have to be present when an application is developed", this feature is not brought out by the claim language. Thus, applicant's arguments are not persuasive.

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7238 for After Final communications, (703) 746-7239 for Official communications and (703) 746-7240 for Non-Official/Draft communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Sue Lao Sue Lao
October 15, 2003